Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Отчет по преддипломной практике

в ООО «Спортмастер»

|  |  |
| --- | --- |
| Студент  Группа АС-17  Руководитель от университета  Руководитель от предприятия  Руководитель ВКР  Доцент, кандидат технических наук | Попов А.Д.    Домашнев П.А.    Власов А.Г.  Назаркин О.А. |
|  |  |

Липецк 2021 г.

ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ АВТОМАТИЗАЦИИ И ИНФОРМАТИКИ

КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

ЗАДАНИЕ НА ПРЕДДИПЛОМНУЮ ПРАКТИКУ

Студенту Попову Антону Дмитриевичу группы АС-17

Направление: 09.03.01 «Информатика и вычислительная техника»

Выполнить:

Ряд задач по frontend разработке на Vue.js

Руководитель практики от предприятия: заместитель директора департамента веб-приложений ООО «Спортмастер» Власов А.Г.

Руководитель практики от ЛГТУ

доцент, к.т.н. Домашнев П.А.

Задание принял к исполнению студент Попов А.Д.

## Аннотация

Данный отчет – это пояснительная записка по пройденному материалу преддипломной практики.

С. 37. Ил. 29. Табл. 8. Лит 3 назв.

Документ включает в себя описание характеристик предприятия и выполнение задания, выданного руководителем практики.

**Оглавление**

[Аннотация 3](#_Toc72426193)

[Введение 5](#_Toc72426194)

[1. Преддипломная практика 6](#_Toc72426195)

[1.1. Ход выполнения 6](#_Toc72426196)

[2. Постановка задачи 21](#_Toc72426197)

[2.1. Литературные и патентный обзор постановки подобных задач 21](#_Toc72426198)

[2.2. Объекты управления, информационные объекты и автоматизируемые процессы. Пользователи и внешние сущности 21](#_Toc72426199)

[2.3. Цели разработки, функции системы, ограничения и критерии оценки результатов 22](#_Toc72426200)

[3. Изучение и моделирование предметной области 24](#_Toc72426201)

[3.1. Выявление основных понятий и процессов, их свойств и закономерностей. Построение ER-диаграммы предметной области 24](#_Toc72426202)

[3.2. Ключевые сценарии, использования системы 29](#_Toc72426203)

[3.3. Теоретическое изучение предметной области. 31](#_Toc72426204)

[4. Разработка информационной базы для решения задачи 33](#_Toc72426205)

[4.1. Построение концептуальной и физической модели данных 33](#_Toc72426206)

[4.2. Спецификация 34](#_Toc72426207)

[5. Список литературы 37](#_Toc72426208)

## Введение

Sportmaster Lab — крупное ИТ-подразделение в рамках международного спортивного ритейлера. 1500 сотрудников обеспечивают информационную поддержку группы компаний Спортмастер.

SM Lab развивает флагманские продукты — продающие сайты и мобильные приложения, разрабатывает внутренние продукты для сотрудников — платформы, сервисы, back-office системы и другое.

Внутри SM Lab используют гибкие практики: Agile, DevOps. Находимся на этапе масштабирования продуктовых команд.

Стек технологий: Java 11, Oracle 19.5, JavaScript, Vue.JS, Spring, Elasticsearch,1С, Kubernetes, Mongo DB, Nginx

## Преддипломная практика

### Ход выполнения

В SM-Lab используется такой инструмент управление разработкой, как канбан-доска (Jira). В порядке очереди задача переходит от аналитиков к back-end, и или front-end разработчикам, если задача общая для одних и для других, то сначала она попадает на back, а далее переходит на фронт, также каждом из этих этапов происходит тестирование. После выполнения и тестирования задачи она попадает на ревью, где коллеги должны оценить правильность и корректность выполнения задачи, если все верно, то задача попадает в prod, если же нет, то нужно исправить замечания и снова отправить на ревью, так пока задача не будет принята.

За время практике мне удалось выполнить 5 задач:

1. Реализовать информационный баннер над хедером. Который будет иметь кнопку «Закрыть». По нажатию на кнопку баннер должен исчезать и состояние должно храниться в local storage, то есть после перезагрузки, баннер не должен появится. Формулировка задачи в JIRA представлена на рисунке 1.

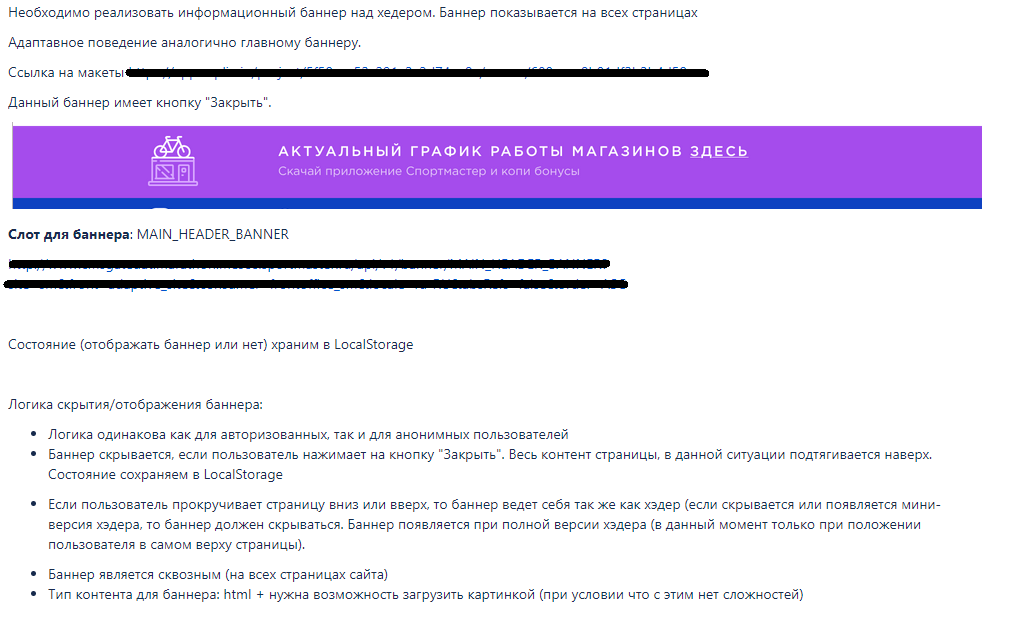


Рисунок 1 - Формулировка задачи 2108 в JIRА

1. Реализовать выбор нужных пунктов в списке, при свернутом списке, выбранные пункты должны отображаться сверху, при развернутом на своих первоначальных местах (как были получены с гейта). Формулировка задачи в JIRA представлена на рисунке 2.

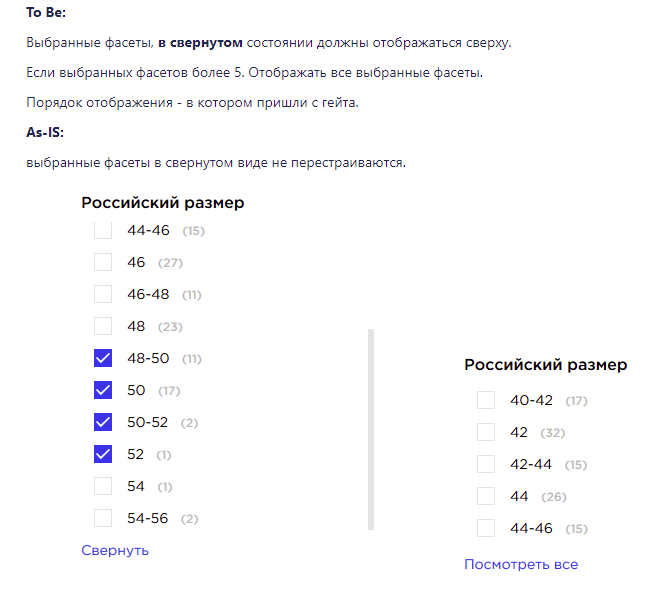


Рисунок 2 - Формулировка задачи 2256 в JIRA

1. Необходимо изменить формат блока «Виды спорта». Изображения должны иметь пропорции 3х4 и не менять их, при изменении размера экрана. Формулировка задачи в JIRA представлена на рисунке 3.

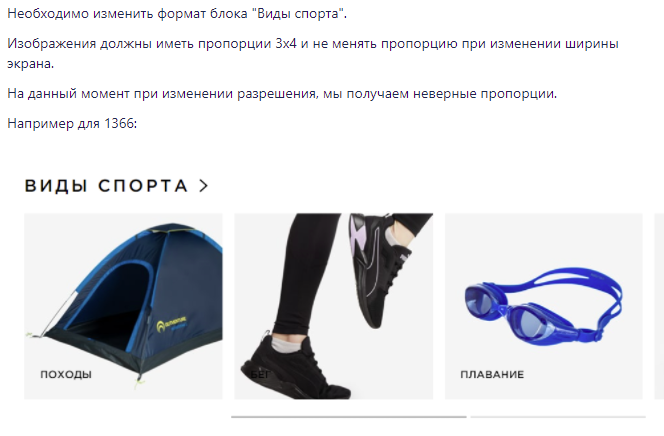


Рисунок 3 - Формулировка задачи 2197 в JIRA

1. Сверстать компонент левого блока футера. Нужно сделать список из трех столбцов. Формулировка задачи в JIRA представлена на рисунке 4



Рисунок 4 - Формулировка задачи 2271 в JIRA

1. Ограничить пользователю применение бонусов, если доступных бонусов для списания 0. Делать недоступным переключатель и показывать соответствующие сообщение. Формулировка задачи в JIRA представлена на рисунке 5.

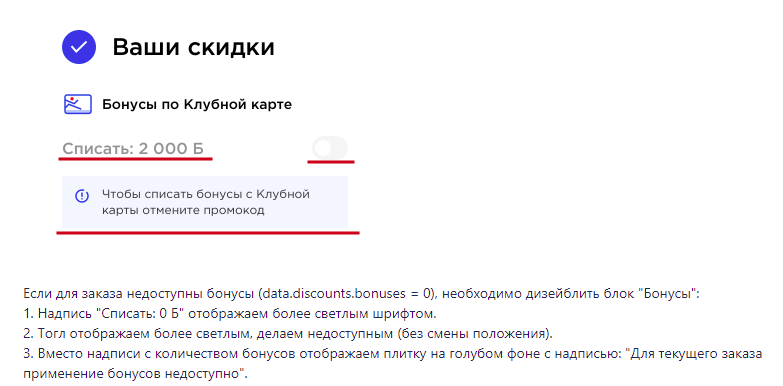


Рисунок 5 - Формулировка задачи 2089 в JIRA

#### Выполнение задачи 2108

Для того чтобы отобразить баннер нужно воспользоваться компонентом SmBanner (специально написанный компонент, который подходит для отображения баннеров). Реализация компонента представлена на рисунке 6.



Рисунок 6 - Компонент SmBanner

Этот компонент принимает следующие значения (props): content – JSON файл, в котором лежит информация url изображения, type – тип контента (image или html). Принимаемые в компонент. Также у компонента есть emit функция errorImage, которая вызывается, если при загрузке баннера произошла ошибка.

Данные для загрузки баннера получаются из гейта с помощью функции getBanner, первым аргументом в которую поступает текущий URL, и название контента для баннера, который нужно получить, в данном случае MAIN\_HEADER\_BANNER. Реализация функции получения баннера (getBanner) представлена на рисунке 7.

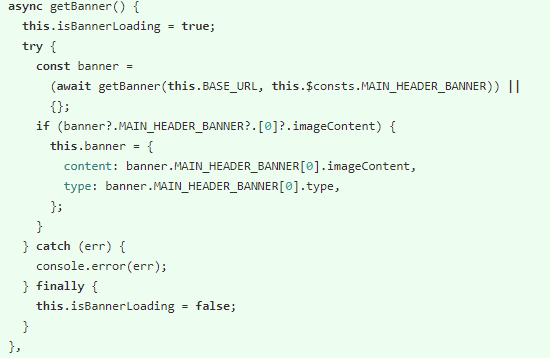


Рисунок 7 - Функция getBanner

Теперь полученные данные можно направить в компонент SmBanner и в результате на странице отобразиться нужный баннер. Баннер изображен на рисунке 8.

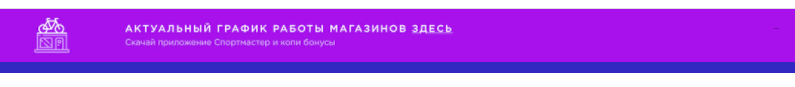


Рисунок 8 - Отображение баннера хедера на странице

Когда баннер добавлен к нему нужно добавить кнопку закрытия и событие, которое будет на эту кнопку реагировать. Реализация кнопки закрытия баннера представлена на рисунке 9.

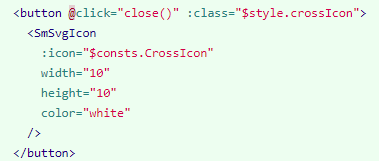


Рисунок 9 - Кнопка закрытия баннера и иконка внутри

Для иконок существует компонент, который принимает изображение, ширину, высоту и цвет, так как иконка формата SVG, у нее можно прямо в верстке менять цвет.

По нажатии на кнопку вызывается функция, которая в свою очередь вызывает emit функцию, в родительском компоненте. Верстка компонента внутри родительского компоненте представлена на рисунке 10, реализация функции, которая проверяет параметр видимости баннера представлена на рисунке 11, а функции, которая проверяет этот параметр на рисунке 12.



Рисунок 10 - Компонент SmHeaderMainBanner в родительском компоненте SmHeader

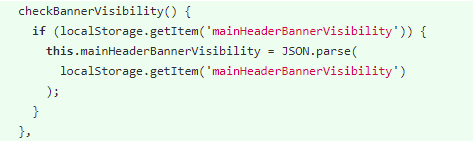


Рисунок 11 - Функция, которая проверяет видимость баннера при загрузки страницы

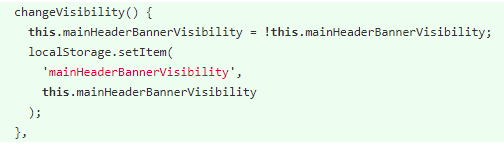


Рисунок 12 - Функция, которая меняет видимость баннера и сохраняет это состояние в LocalStorage

На рисунке 13 представлено локальное хранилище до того, как баннер на хедер был закрыт, а на рисунке 14 после этого.

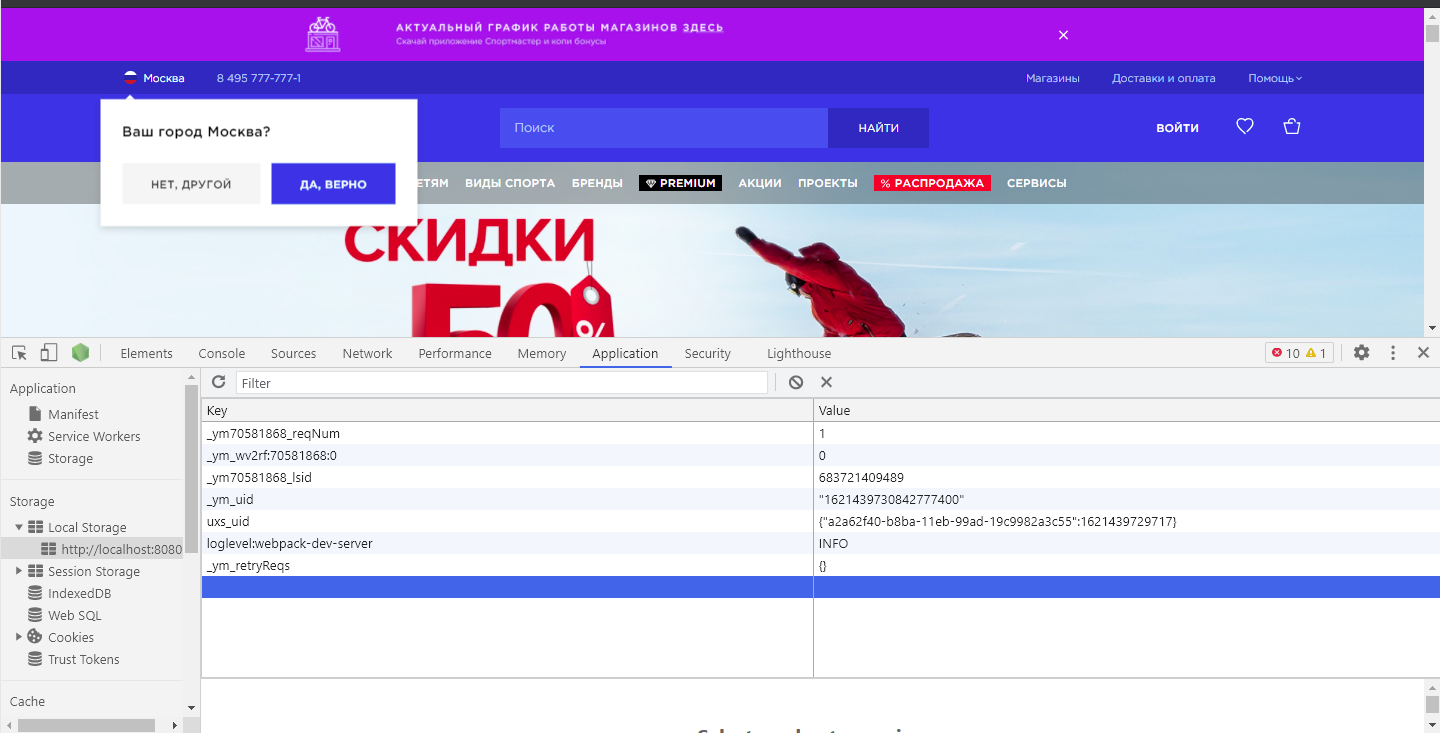


Рисунок 13 - Страница и LocalStorage, до нажатия на "крестик"

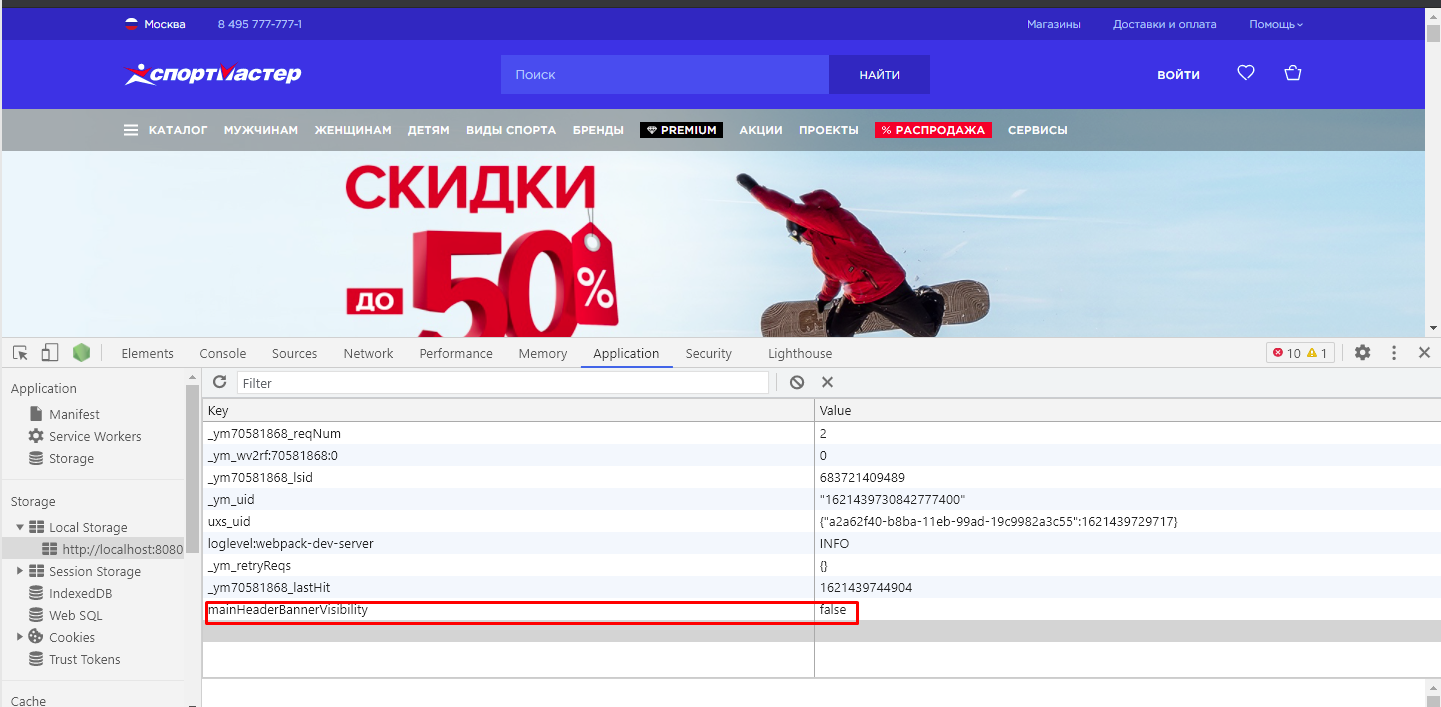


Рисунок 14 - Страница и LocalStorage, после нажатия на "крестик"

После нажатия на кнопку «закрыть» в локальном хранилище появилась запись о том, что баннер скрыт

#### Выполнение задачи 2256

Для того, чтобы отображать выделенные категории над невыделенными, нужно отсортировать массив. Создается два пустых массива (selected и unselected), когда встречается элемент, у которого значение selectedByUser равно true, он отправляется в массив selected, если false, то в unselected.

Но когда список раскрыт, то не нужно отображать отсортированный массив, поэтому нужно еще проверить значение showAllValues, которое равно true, когда список раскрыт, и false, когда закрыт. На рисунке 15 представлена реализация функции, которая сортирует массив по параметру selectedByUser.

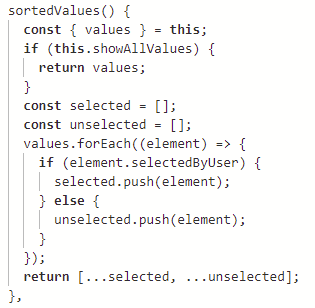


Рисунок 15 - Функция сортировки массива

На рисунке 16 представлен свернутый список, на котором выделенные категории находятся над невыделенными. На рисунке 17, видно, когда список раскрыт, то категории находятся в изначальном порядке.



Рисунок 16 - Скрытый список

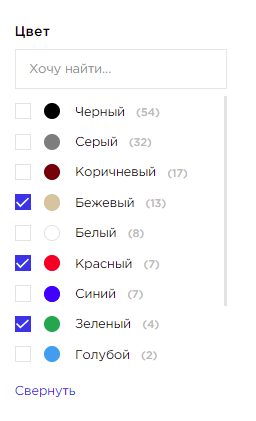


Рисунок 17 - Раскрытый список

#### Выполнение задачи 2197

Очень простая задача, где нужно было немного подредактировать scss файл.

У блока с картинкой нужно было убрать все ограничения по высоте, поставить ее автоматической и в готовой scss функции прописать нужное соотношение сторон. Эти действия представлены на рисунке 18.



Рисунок 18 - Реализация задачи 2197

#### Выполнение задачи 2271

Для отображения списка в футере был создан новый компонент SmFooterLeftBlock, входными параметрами будут данные, полученные из стейта.

В этом компоненте перебирается массив, который делится на колонки, который в свою очередь делится на секции. Элементы секций могут быть ссылками, а могут быть просто текстом, если у элемента поле с url пустое, или его просто нет, то отображать как текст, если есть, то отображать ссылку. На рисунке 19 представлена реализация верстки компонента SmFooterLeftBlock, на рисунке 20 его стилизация.



Рисунок 19 - Верстка компонента SmFooterLeftBlock



Рисунок 20 - Стилизация компонента

На рисунке 21 представлен готовых компонент на странице сайта.

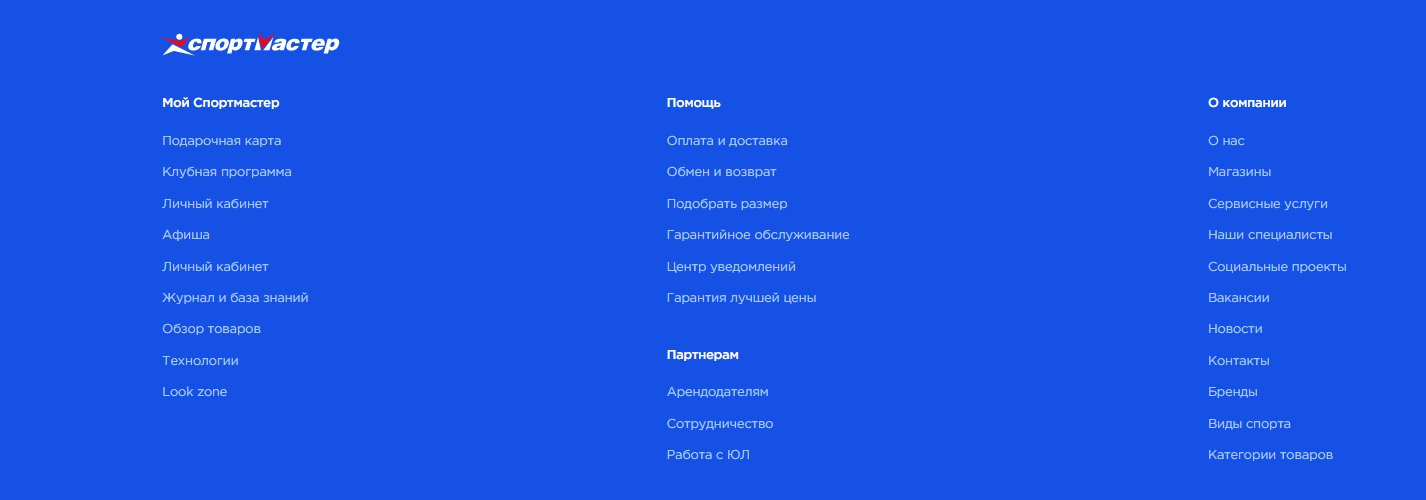


Рисунок 21 - Компоненты SmFooterLeftBlock

#### Выполнение задачи 2089

В этой задаче нужно вывести сообщение, если применение бонусов недоступно. Для отображения сообщения используем заранее написанный компонент SmMessage. На рисунке 22 представлена верстка компонента.



Рисунок 22 - Компонент SmMessage

Для того, чтобы компонент показывался только тогда, когда нам нужно, используется надстройка v-if, если значение, которое она принимает истинное, то компонент показывается, если нет, то нет. Значение для v-if используем с помощью функции disabledToggle, которая возвращает true, если есть бонусы, и false, если нет

Еще нужно сделать недоступный переключатель применения бонусов, у него есть поле disabled, которое отвечает за то, чтобы переключатель был активным или неактивным. В это поле также записывается значение из disabledToggle. На рисунках 23 и 24 представлены изображения компонентов, когда можно воспользоваться бонусами, и когда нельзя.

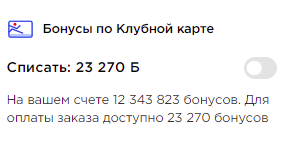


Рисунок 23 - Возможно списать бонусы



Рисунок 24 - Невозможно списать бонусы

## Постановка задачи

### Литературные и патентный обзор постановки подобных задач

На данный момент на рынке представлено несколько решений, которые помогают пользователю в обучении, но все они не являются в достаточной мере удобными. Тем более, если нужно обучить языку ребенка, то серое и невзрачное приложение вряд ли сможет его завлечь. Также нет готового решения, которое было бы представлено в виде веб-приложения, то есть программного продукта, который возможно открыть как на компьютере, смартфоне, так и на любом другом устройстве с доступом в интернет.

Ниже представлены некоторые примеры систем для изучения жестового языка для android:

* «Русский жестовый Язык», «Словарь РЖЯ-112». Это два приложения с похожим функционалом. Пользователю показывают ряд изображений с комментариями для изучения букв, слов, после чего ему предлагается закрепить материал, то есть демонстрируют изображение с какой-то определенной буквой или словом и пользователю нужно выбрать правильный ответ.
* «Методы изучения жестового языка». Данное приложение предлагает пользователю набор картинок с изображением жеста и того, что он означает.

### Объекты управления, информационные объекты и автоматизируемые процессы. Пользователи и внешние сущности

Объектом управления является процесс обучения

Автоматизируемые процессы:

* отслеживание ошибок в обучении
* ведении статистики успешности прохождения материала

Пользователи системы:

1. Основной пользователь – человек с ограниченными возможностями:
2. прохождение уроков
3. изучение теоретических материалов
4. создание индивидуальной коллекции упражнений
5. регистрация и авторизация в системе
6. просмотр статистики успешности прохождения материала
7. Администратор:
8. редактирование списка упражнений
9. редактирование списка теоретических материалов
10. авторизация в системе

Человек является внешней сущностью системы. Так как он взаимодействует с системой, то есть он является источником входной информации (администратор) и также является приёмником выходной информации (основной пользователь)

### Цели разработки, функции системы, ограничения и критерии оценки результатов

#### Цели разработки

Целями создания системы являются:

* упрощение процесса изучения жестового языка
* отслеживание ошибок в обучении
* отслеживание статистики успешности прохождения материала

#### Функции системы

* Обеспечение авторизированного доступа в систему
* Отображение инструкции по взаимодействию с системой
* Предоставление администратору возможности редактировать коллекции обучающих материалов
* Предоставление пользователю упражнений для изучения жестового языка
* Информирование пользователя об успешности пройденных им упражнения или тестирования (отображение оценки)
* Предоставление пользователю теоретических сведений
* Предоставление пользователю статистики прохождения упражнений

#### Критерии оценки эффективности

* является ли интерфейс удобным и интуитивно понятным.
* смогла ли система сократить время, потраченное на изучение РЖЯ
* корректное функционирование системы

#### Ограничения

Главным ограничением работы системы является наличие интернет соединения

## Изучение и моделирование предметной области

### Выявление основных понятий и процессов, их свойств и закономерностей. Построение ER-диаграммы предметной области

#### Архитектура системы

Перед тем как реализовывать выбранную систему, нужно определиться с выбором платформы, для которой будет она создаваться. Так как очень важно, чтобы обучающую система была кроссплатформенная, выбор пал на реализацию системы в виде веб-приложения с использованием языка JavaScript.

Далее нужно выбрать архитектуру разрабатываемого продукта. Выбор пал на REST архитектуру, которая позволит поддерживать высокую производительность системы и простоту взаимодействия между клиентом и базой данных. Приложение будет работать по принципу клиент-сервер. Взаимодействие с API будет осуществляться по средствам HTTP протокола с помощью методов GET (получение данных), POST (добавление данных), PUT (изменение данных) и DELETE (удаление данных). Для передачи данных будет использоваться формат JSON.

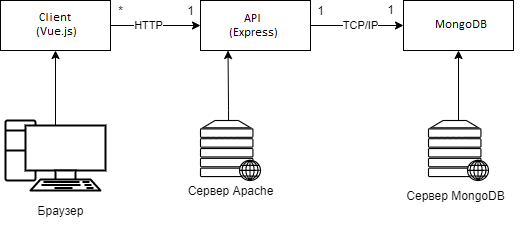


Рисунок 25 - Диаграмма развертывания

#### Основные понятия

1. Жестовый язык — самостоятельный язык, состоящий из жестов, каждый из которых производится руками в сочетании с мимикой, формой или движением рта и губ, а также в сочетании с положением корпуса тела. Эти языки в основном используются в культуре глухих и слабослышащих с целью коммуникации. Использование жестовых языков людьми без нарушения слуха вторично, однако довольно распространено: часто возникает потребность в общении с людьми с нарушениями слуха, являющимися пользователями жестового языка.
2. Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети
3. Основной пользователь – человек, который использует систему для изучения жестового языка
4. Администратор – человек, который добавляет в систему новые данные и редактирует уже имеющиеся
5. Упражнение – совокупность определенных жестов и соответствующих им теоретических материалов
6. Тестирование – совокупность ранее пройденных пользователем жестов, без демонстрации теоретических материалов.

#### Диаграмма вариантов использования

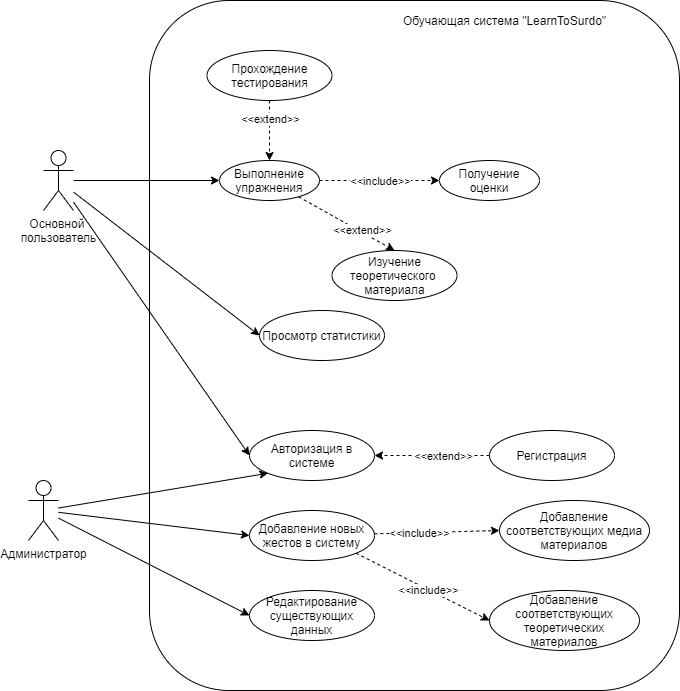


Рисунок 26 - Диаграмма вариантов использования

Основной пользователь:

* Регистрируется в системе
* Авторизируется в системе
* Выполняет упражнения или тестирование
* Получает оценку по завершении упражнения
* Изучает теоретические материалы
* Просматривает статистику успешности прохождения уроков.

Администратор:

* Авторизируется в системе
* Добавляет новые данные в систему
* Редактирует уже существующие.

#### ER-диаграмма предметной области

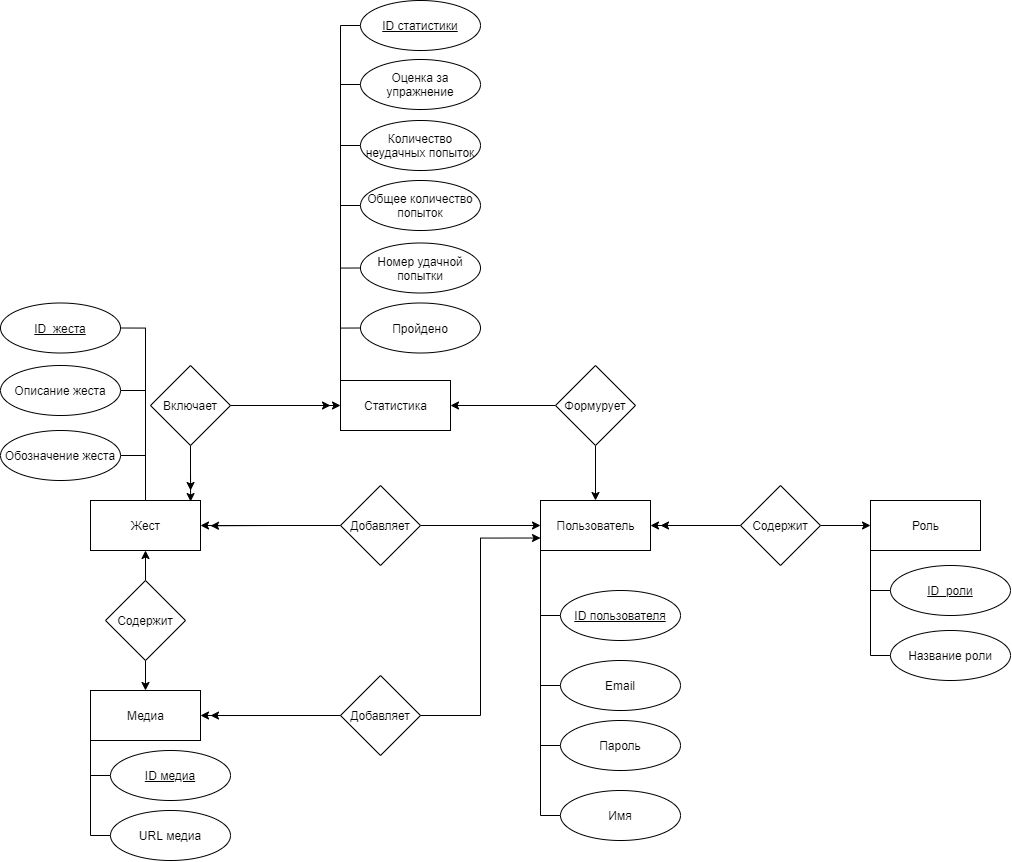


Рисунок 27 - ER-диаграмма в нотации Чена

Таблица 1 - Спецификация сущностей

|  |  |
| --- | --- |
| Имя сущности | Имя атрибута |
| Пользователь | ID пользователя |
| Email |
| Пароль |
| Имя |
| Статистика | ID статистики |
| Оценка за упражнение |
| Количество неудачных попыток |
| Общее количество попыток |
| Номер удачной попытки |
| Пройдено |
| Жест | ID жеста |
| Описание жеста |
| Обозначение жеста |
| Медиа | ID медиа |
| URL медиа |
| Роль | ID роли |
| Название роли |

1. Пользователь: Это или простой человек, который регистрируется в системе. В этой сущности будут хранится данные для авторизации пользователя, его идентификатор, email, пароль в зашифрованном виде и имя, или это администратор: человек, который занимается редактированием данных в системе, удалением и добавлением. Добавляет в систему новые жесты и соответствующие им медиа материалы
2. Статистика: сущность, которая будет связана с определенным жестом. В ней будет хранится то, сколько раз человек неудачно выполнил задание с выбранным жестом, то есть не смог набрать за прохождение самую низкую оценку. Так же здесь же фиксируется оценка за прохождение и флаг указывающий на то, пройдено задание или нет. Фиксируется общее количество попыток, и номер попытки, после которой пользователь смог выполнить задание, хотя бы на минимальную оценку
3. Жест: сущность, в которой хранится описание жеста, который связан с сущностью медиа.
4. Медиа: сущность, в которой находится URL до картинки или гифки жеста.
5. Роль: сущности, в которой будет находится идентификатор роли и ее название.

### Ключевые сценарии, использования системы

#### Основной сценарий взаимодействия пользователя с системой

1. Пользователь авторизуется или регистрируется в системе.
2. Пользователя переадресовывают на главную страницу.
3. Пользователь выбирает страницу, на которую хочет перейти (упражнения, тестирование, статистика, профиль).
4. Пользователь взаимодействует со страницой.
5. Пользователь выходит из системы.
6. Конец.

#### Сценарий взаимодействия пользователя со страницей упражнения

1. Пользователь переадресовывается на страницу «Упражнения»
2. Пользователь выбирает категорию, которую хочет изучить.
3. Пользователю демонстрируется до 5 теоретических материалов, из числа тех, которые не были пройдены ранее.
4. После изучения теоретического материала пользователь выполняет до 5 заданий на закрепление пройденного материала.
5. Если пользователь выполнил все верно, то ему демонстрируется следующие 5 упражнений, если нет, то еще раз проходит те, что выполнил неверно.
6. Конец.

#### Сценарий взаимодействия пользователя со страницей тестирование

1. Пользователь переадресовывается на страницы с тестированием.
2. Пользователю предлагается выполнить от 10 до 20 заданий, которые были выбраны случайным образом из ранее пройденных.
3. Пользователь выполняет задания.
4. Пользователю показывают количество правильных и неправильных ответов, а также те задания, в которых он допустил ошибку.
5. Конец.

#### Сценарий взаимодействия пользователя со страницей статистика

1. Пользователю переадресовывается на страницу со статистикой.
2. Пользователю демонстрируют список жестов, которые он уже прошел, и все статистическую информацию о них.
3. Конец.

#### Сценарий взаимодействия пользователя со страницей профиль

1. Пользователь переадресовывается на страницу своего профиля.
2. Пользователь может изменить пароль, или имя.
3. Конец.

#### Сценарий добавления новых жестов

1. Пользователь авторизуется в системе как администратор
2. Пользователь вводит в соответствующие поля название жеста и описание. Выбирает категорию, к которой жест относится.
3. Нажимает кнопку отправить «Конец»
4. Конец.

#### Сценарий добавления новых категорий

1. Пользователь авторизуется в системе как администратор
2. Пользователь вводится в соответствующие поле название категории.
3. Нажимает кнопку «Отправить»
4. Конец.

#### Сценарий редактирования данных

1. Пользователь авторизуется в системе как администратор.
2. Выбирает из таблицы поле, которое нужно изменить.
3. Меняет данные и сохраняет.
4. Конец.

### Теоретическое изучение предметной области.

В приложении существует два варианта изучения жестового языка:

* Упражнения
* Тестирование

#### Упражнения

В упражнениях пользователю предлагается выбрать одну из категорий, слова в которой, он хочет изучать. Существуют следующие категории:

* Алфавит
* Числа
* Знакомство
* Человек
* Родство, семья
* Распространенные фразы

Каждая из этих категорий содержит в себе теоретические материалы, в которых описывается жест, а также демонстрируется его изображение, если жест статичный, или гиф, если жест происходит в движении.

В упражнениях пользователь сначала изучает теорию, а после закрепляет ее на практике.

#### Тестирование

Когда пользователь выполнил хотя бы 10 упражнений, ему становится доступна категория тестирование, где пользователю случайным образом демонстрируется набор ранее пройденных жестов. После прохождения тестирование пользователь видит результат, количество верных и неверных ответов, а также те жесты, которые были отвечены неверно.

## Разработка информационной базы для решения задачи

### Построение концептуальной и физической модели данных

#### Концептуальная модель данных



Рисунок 28 - Концептуальная модель данных

#### Физическая модель данных



Рисунок 29 - Физическая модель данных

### Спецификация

#### Спецификация сущностей

Таблица 2 - Спецификация сущности «Роли»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Расшифровка | Тип данных | Обязательность | Первичный ключ |
| Role\_ID | Идентификатор роли | Integer | + | + |
| Role\_Title | Название роли | VarChar(255) | + | - |

Таблица 3 - Спецификация сущности «Пользователи»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Расшифровка | Тип данных | Обязательность | Первичный ключ |
| User\_ID | Идентификатор пользователя | Integer | + | + |
| User\_name | Имя пользователя | VarChar(255) | + | - |
| Email | Электронная почта пользователя | VarChar(255) | + | - |
| Password | Пароль пользователя | VarChar(255 | + | - |

Таблица 4 - Спецификация сущности «Жесты»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Расшифровка | Тип данных | Обязательность | Первичный ключ |
| Gesture\_ID | Идентификатор жеста | Integer | + | + |
| Gesture\_Title | Название жеста | VarChar(255) | + | - |
| Gesture\_Description | Описание жеста | Text | + | - |

Таблица 5 - Спецификация сущности «Категории»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Расшифровка | Тип данных | Обязательность | Первичный ключ |
| Category\_ID | Идентификатор категории | Integer | + | + |
| Category\_Title | Название категории | VarChar(255) | + | + |

Таблица 6 - Спецификация сущности «Медиа»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Расшифровка | Тип данных | Обязательность | Первичный ключ |
| Media\_ID | Идентификатор медиа | Integer | + | + |
| Media\_URL | Путь, по которому хранится медиа файл | VarChar(500) | + | - |

Таблица 7 - Спецификация сущности «Статистика»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Расшифровка | Тип данных | Обязательность | Первичный ключ |
| Statistics\_ID | Идентификатор статистики | Integer | + | + |
| Completed | Флаг выполнения задания | Boolean | + | - |
| Num\_of\_Unsuccessful\_Attempts | Количество неудачных попыток | Integer | + | - |
| Num\_of\_Attempts | Общее количество попыток | Integer | + | - |
| Successful\_Attempt\_Num | Номер удачной попытки | Integer | + | - |

#### Спецификация связей

Таблица 8 - Спецификация связей

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Название | Сущность 1 | Сущность 2 | Обязательность 1 | Обязательность 2 | 1,2 | 2,1 |
| Contains\_Role | Users | Roles | + | + | n:1 | 1:n |
| Creates\_Media | Users | Media | + | - | 1:n | n:1 |
| Creates\_Categories | Users | Categories | + | - | 1:n | n:1 |
| Creates\_Gestures | Users | Gestures | + | - | 1:n | n:1 |
| Keeps | Users | Statistics | + | - | 1:n | n:1 |
| Constains\_Statistrics | Gestures | Statistics | + | - | 1:n | n:1 |
| Includes | Gestures | Categories | - | + | n:1 | 1:n |
| Contains\_Media | Gestures | Media | - | + | n:1 | n:1 |

## Список литературы

1. vuejs.org/v2/guide/ [Электронный ресурс]: Документация по Vue.js
2. expressjs.com/ru/ [Электронный ресурс]: Документация по Express.js
3. JavaScript. Полное руководстов. 7-е издание. Дэвид Флэнаган [Книга]: Справочник по JavaScript